Teaching Philosophy 42:1, March 2019
DOI: 10.5840/teachphil2019116101

Two-Sided Trees for Sentential Logic, Predicate Logic, and Sentential Modal Logic

JESSE FITTS
DAVID BEISECKER
University of Nevada Las Vegas

Abstract: This paper will present two contributions to teaching introductory logic. The first contribution is an alternative tree proof method that differs from the traditional one-sided tree method. The second contribution combines this tree system with an index system to produce a user-friendly tree method for sentential modal logic.

1. Introduction

This paper will present two contributions to teaching introductory logic. The first contribution is an alternative tree proof method that differs from the traditional one-sided tree method from Smullyan 1968 and popularized in Jeffrey 1967.[1] Jeffrey initially presents the tree method as a more efficient means of searching for counterexamples, which is a semantic endeavor.[2] The tree system properly understood, however, is not merely a more efficient semantic mechanism for producing counterexamples but rather a fully sound and complete syntactic proof system for sentential logic (SL) and predicate logic (PL).

Trees, both traditional and the alternative we present below, differ from other popular proof systems such as natural deduction in various ways, some of which may be advantageous depending on the professor's aims. With trees, there are not different proof strategies, for example direct or indirect. At least in the sentential setting, the order of rule application doesn't matter to the success of the proof, though in some cases it may count for or against its elegance. There is only one rule to deploy on each well-formed formula (wff). The tree system doesn't require the student to build a strategic sense to look ahead in the proof to craft a strategy. If a professor is interested in building such a strategic sense, then natural deduction would be a natural proof system. The tree system might seem "mindless," but if the instructor is interested in featuring the syntax-semantics distinction, this is precisely what is needed, especially as more computer science students enroll in our courses. The instructor could explain how it would be relatively straightforward to program a mindless computer to deploy the tree system (at least for SL) while it might not be so straightforward to do the same with natural deduction.

The alternative tree system we present in this paper accrues the benefits of the traditional tree system, including soundness and completeness, yet adds more. Very roughly, the basic differentiating feature of this system is that a diagrammatic feature of trees—the regions of *right* and *left*—play a crucial role. In particular, the right-hand side of tree paths plays a role that is played by the negation sign in more traditional trees—that of denying a claim. That the system relies on the semantically indeterminate notions of left and right helps reinforce the syntax-semantics distinction. We present the system and further explain its merits in §2.

The second contribution of this paper, presented in §3, is a new tree system for propositional modal logic. This system combines the just-described two-sided tree system with a way of keeping track of the accessibility relation on the tree itself—without arrows or an annotation column—with *indices*. Gary Hardegree, in an unpublished modal logic textbook that employs natural deduction proofs,[3] first proposed, as far as we are aware, the index system we employ. We modify the index system and deploy it in the context of trees. The main innovation of this section of the paper will be this index system combined with trees—the index system doesn't necessarily have to combine with two-sided trees.

2. Two-sided Trees

2.1 Two-Sided Trees for SL and PL

We'll be testing whether the expression $\ulcorner \Gamma \vdash \varphi \urcorner$ is correct, where Γ is a set of wffs, and φ is a single wff.[4]

Here is the method for constructing a two-sided tree. To start, we place the wffs in Γ on the left and φ on the right of a line. Here, note that adding outside parentheses and negating φ is not required as with traditional trees.

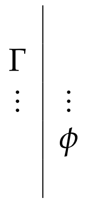


Figure 1

We develop a tree by applying the rules stated below, some of which require branching and others which do not. A path of a tree closes when the same wff appears on both the left and right of a path. We then close that path, without any further development, and place an '×' below it:

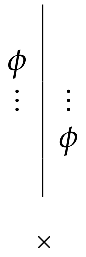


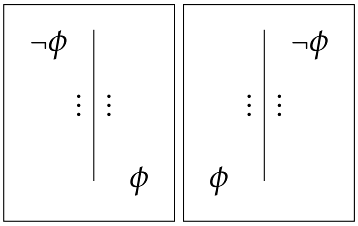
Figure 2

The vertical ellipses represent various wffs and branching that can intervene between φ on the left and right.

- **Teaching note**: We tell the students that if they have φ on one side and can trace from φ only upwards or sideways (without picking up the pencil) and find φ on the other side, that branch closes.

Here are the rules, where lower-case, italicized, Greek letters range over arbitrary wffs:

Negation



Branching

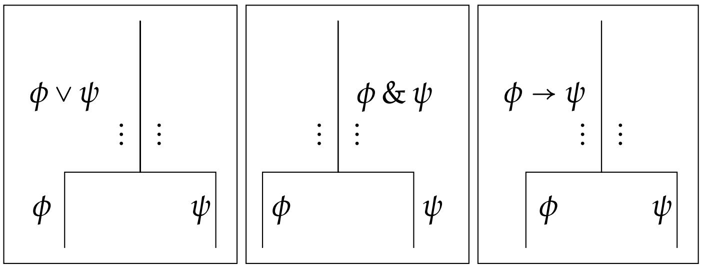


Figure 3a

Non-branching

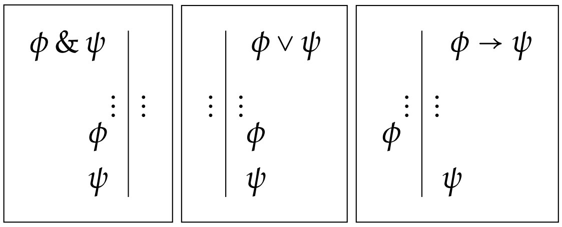


Figure 3b

A tree path is fully developed if and only if it is closed or no more rules can be applied without duplication. An entire tree is fully developed just in case every path is fully developed. If every path closes, we have a closed tree and thus $\Gamma \vdash \varphi$. If the tree is fully developed and at least one path is open, then $\Gamma \nvdash \varphi$.

- **Teaching note**: As with traditional trees, to close paths efficiently, we apply non-branching rules before branching.

2.1.1 Merits of the System

Here are a few merits of the just-presented system:

The system is fully decompositional. With two-sided trees, every application of a rule only results in wffs that are strictly smaller than, in terms of the number of operators, the wff to which the rule was applied. This differs from the traditional tree system. For example, in that system, when we apply the conditional rule, on one branch, we add a negation:

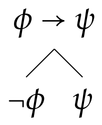


Figure 4

Two-sided trees have the separation property. Two-sided trees have a property referred to in philosophical logic as *separation*, the property of a proof system such that a proof of a wff only involves the operators involved in that wff.[5] This property is related to the fully decompositional point: as the reader can verify, the rules for any given wff don't involve operators not in that wff. This is unlike, as we've just seen, the rule for the conditional in traditional trees, which involves negation. Briefly, the reason that this is important, as Humberstone (2000: 357–58) explains, is that, given that in this context validity and provability go hand in hand, there will be tension between (a) the plausible view that the rules governing connectives exhaust their meanings, and that (b) validity is a function of logical vocabulary. A wff containing only some set of connectives, say just the conditional in the theorem ⌜ $((\varphi \to \psi) \to \varphi) \to \varphi$ ⌝ (Peirce's Law) will only be provable with appeal to the rules for '→' and '¬' with a traditional tree. But then, Peirce's Law isn't valid solely in virtue of the conditional if the meaning of the conditional is determined by negation as well.

Relatedly, there are no "derived" rules in the two-sided tree system. For example, in the traditional tree system we need rules to decompose operators that negation govern. For example:

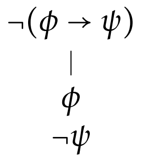


Figure 5

In the two-sided system, in the equivalent situation, we simply move the wff that the negation governs, i.e., ⌜ $\varphi \to \psi$ ⌝, from the left side to the right side and then deploy the arrow-on-right rule:

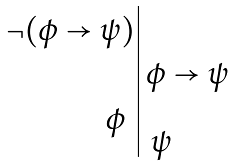


Figure 6

Scheffer and Peirce Strokes don't require negation.[6] Relatedly again, given the full decompositionality, we can add rules for the Peirce and Scheffer Strokes to the two-sided tree system that don't require negation to state the rules. The rules for the Scheffer (up-arrow) and Peirce (down-arrow) strokes in the traditional tree system would look, respectively, as follows:

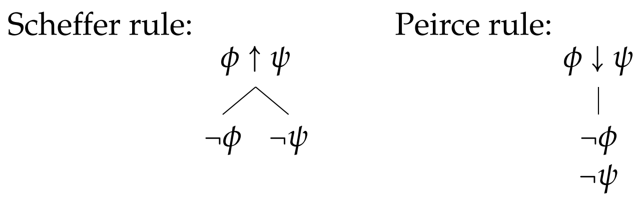


Figure 7

It is impossible to state the rules for these strokes in the traditional tree system without recourse to negation in the statement of the rule.

The two-sided tree rules for Scheffer and Peirce look as follows: Scheffer:

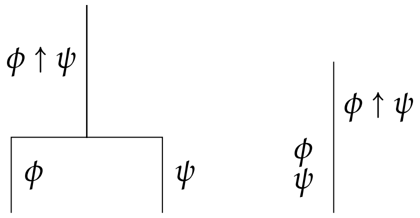


Figure 8

Peirce:

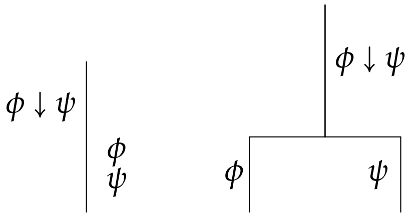


Figure 9

- **Teaching note**: A group project with which one of the authors has had great success is, after the class is familiar with the two-sided tree system (this also works for one-sided trees), to have the students come up with the appropriate rules for the above operators from their truth tables. This also works for other operators, such as the biconditional, or even exotic operators, such as the operator ⌜ $\varphi \nleftarrow \psi$ ⌝, which has the same truth table as ⌜ $\neg(\psi \rightarrow \varphi)$ ⌝.

These two strokes may seem like arcane symbols, but teaching them, especially in an upper-division setting, is useful for a couple of reasons. First, many computer science students will be familiar with the strokes via their study of logic gates. Second, and more important, is the issue of expressive completeness. A common way to begin an advanced logic course is by first investigating the idea of an expressively complete set of connectives—a set of connectives that can represent any truth function. We show with conjunctive and disjunctive normal forms that we can express every truth function with {'¬', '∨', '&'}, and then we show with de Morgan's laws that just {'¬', '∨'} or {'¬', '&'} will suffice. Finally we surprise the class by telling them that just the Scheffer or just the Peirce stroke is expressively complete. We could, we tell the class, just get by with only one of them. But once we get to the syntactic proof section of the course, this doesn't turn out to be true on a traditional tree system since it is impossible to state the strokes' rules without recourse to negation. Furthermore, a comment that we've made during the expressive completeness section is that the more parsimonious the set of connectives in our system, the simpler the meta-theory. In particular, the soundness proof for any tree system proceeds via clauses for each rule in one's system, and that, with the expressive completeness of the strokes, we could have just one rule justification. But this isn't right since those rules' statements make reference to extraneous operators, viz., negation.

The system again reinforces the syntax-semantics distinction. Finally, we wish to reinforce the idea that the two-sided tree system helps the instructor focus on the syntax-semantics distinction. This system again employs the semantically indeterminate notions of right and left, and we could easily rebuild an alternative system by switching the roles. We can teach the system without any reference to the meanings of the connectives, and this being the case, we could teach the system for both SL and PL before teaching their respective semantic sections.

2.1.2 Illustrative Problems

(1) {'*P*', '*P* → *Q*'} ⊢ '*Q*':

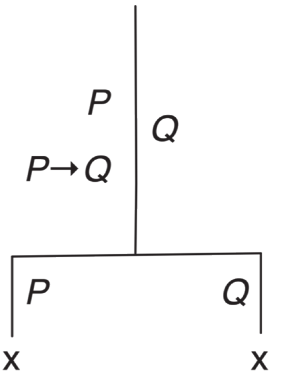


Figure 10

(2) {'¬*P* ∨ *Q*', '¬*Q*', '*R* → *P*'} ⊢ '¬*R*'

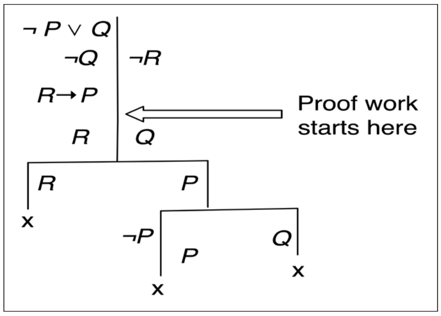


Figure 11

(3) $\vdash$ ‘$(A \rightarrow B) \vee (B \rightarrow C)$’

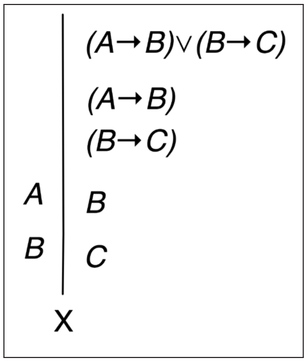


Figure 12

Notice that no indirect strategies are needed here—just three steps corresponding to each major connective.

2.2 Two-Sided Trees for PL

For PL, we introduce four new rules, two for each quantifier. Where lowercase, italicized, Greek variables range over arbitrary wffs, a bolded ‘**x**’ is a metalinguistic, schematic letter for object-level variables, and where ‘α’ is an individual constant:

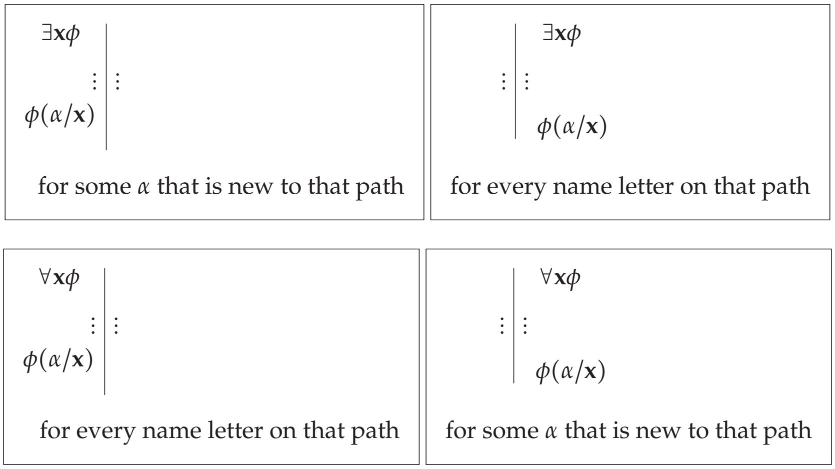

∃xϕ
ϕ(α/x)
for some α that is new to that path
∃xϕ
ϕ(α/x)
for every name letter on that path
∀xϕ
ϕ(α/x)
for every name letter on that path
∀xϕ
ϕ(α/x)
for some α that is new to that path

Figure 13

A few notes: For universal on the left and existential on the right, if there are no names on that path, use any name. Universals on the left and existentials on the right can be reused. These rules are stated in terms of fully developing a tree. When we’re doing proofs to show that some single turnstile claim holds, we strategically instantiate universals on left and existentials on right after performing the other two rules, when relevant.

2.2.1 Illustrative Problems

(4) '∃*x*(*Fx* ∨ *Gx*)' ⊢ '∃*xFx* ∨ ∃*xGx*'

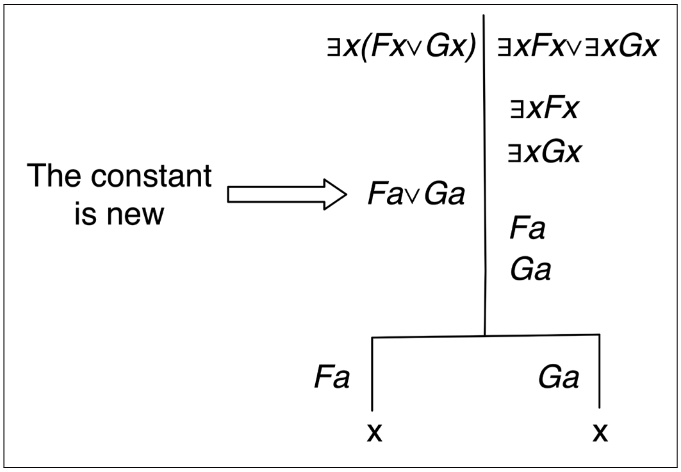


Figure 14

(5) '∀*y*∃*x*(*Fx* ∨ *Gy*)' ⊢ '∃*x*∀*y*(*Fx* ∨ *Gy*)'

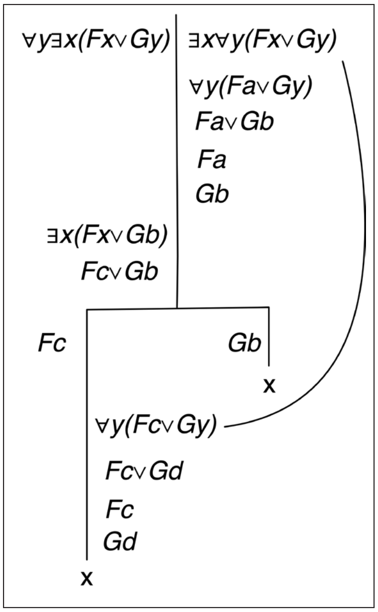


Figure 15

The line indicates that '∃*x*∀*y*(*Fx* ∨ *Gy*)' has been reused to close a branch.

3. Two-Sided Trees for Sentential Modal Logic (SML)

This section presents, with an aim at popularizing, a way of keeping track of accessibility relations on a tree with an index system due to Gary Hardegree. This system has a few merits:

- Indices are fully syntactic objects. Because of this, there is no aspect of the proof system that contains semantic information. This is unlike the popular system of Girle 2000, which contains an annotation column that contains semantic information—i.e., it explicitly keeps track of accessibility relations. Again, this gives the instructor another opportunity to stress the syntax-semantics distinction, and the rules for manipulating indices are couched in terms of concatenation rules that computer science students are often familiar with.
- The index system differs in some ways from the arrow system of another popular modal logic textbook—Garson 2006. Garson's system visually makes clear the various accessibility relations, and it may be easier to recover such facts while looking at a student's proofs. The index system, on the other hand, requires less semantic thought while in the midst of the proof, and it is clearer how one could write a computer program for the index-tree system.

Two-sided SML trees will resemble two-sided trees for SL, with the addition of modal operators. In addition to this, we append an index to every wff on the tree that represents the world at which that wff is evaluated. We'll start by introducing the index system.

3.1 The Index System and Different Modal Systems

An index is a sequence of numerals that is a compound name for a possible world. We always start every wff on the tree from the index 0, we don't skip numerals, and every index ends with a unique numeral. As we'll see below, the rules for '$\Diamond$' on the left and '$\Box$' on the right always generate new worlds. So for example, if we're at 0 and we use $\Diamond$-out on the left on $\ulcorner \Diamond\varphi \urcorner$, and 1 is available, then we place φ at 01 on the left. The system is best introduced via example. Below is a sequence of indices displayed as a tree. I use a tree because the system is similar to a matronymic/patronymic naming system.

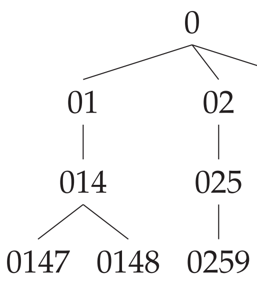


Figure 16

We have a principle that governs accessibility for every system of modal logic:

Tree accessibility principle: When one world generates another, the generated world is accessible from the generating world.

The analogue in terms of indices is as follows:

Index dependency and accessibility: If *m* is a decedent from *i*, then m is accessible from *i*.

So, e.g., 0147 is accessible from 014, which is accessible from 01, which is accessible from 0.[7] We can then specify the historically important modal systems by giving restrictions on indices. Each of these systems comes with a characteristic thesis. Here are those important systems along with the characteristic thesis of each system.

System	Restriction on accessibility	Characteristic thesis
K	No restriction	$\Box(\phi \to \psi) \to (\Box\phi \to \Box\psi)$
D	Serial	$\Box\phi \to \Diamond\phi$
T	Reflexive	$\Box\phi \to \phi$
B	Reflexive and Symmetric	$\Diamond\Box\phi \to \phi$
S4	Reflexive and Transitive	$\Box\phi \to \Box\Box\phi$
S5	Reflexive, Symmetric, and Transitive	$\Diamond\Box\phi \to \Box\phi$

Figure 17

We're now in a position to state the rules for SML.

3.2 Rules for Two-Sided SML trees

We update our closure rule so that a wff on the left and right at the same world closes a branch. Where a bold '**w**' ranges over indices:

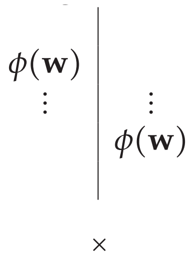


Figure 18

The SL rules will stay the same with a world index appended to each wff. E.g., if you have ⌜ ¬φ ⌝ at **w** on the left—i.e., ⌜ ¬φ(**w**) ⌝—you move ⌜ φ(**w**) ⌝ to the right to deploy the negation-on-left rule. Where "new" refers to a new numeral, here are the rules for the modal operators (where a bold '**v**' also ranges over indices):

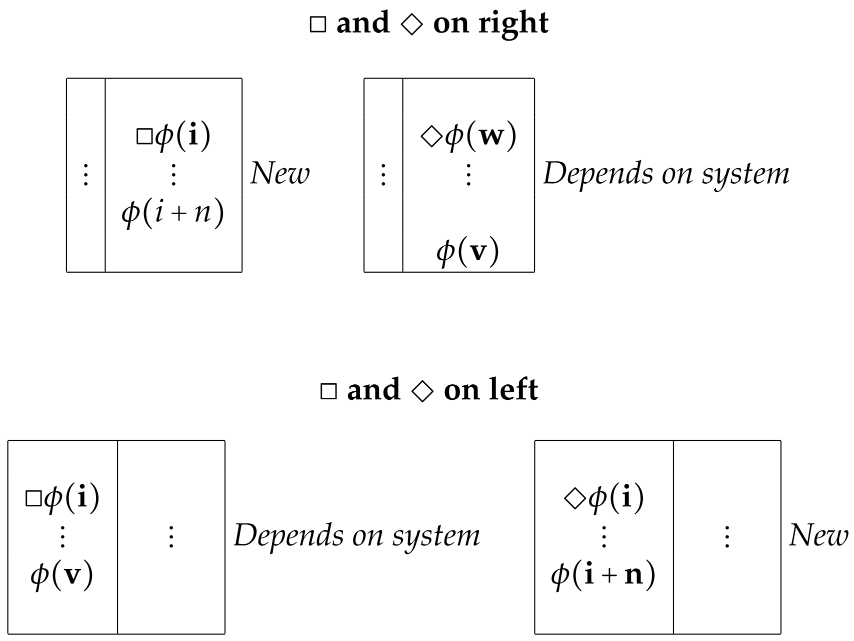


Figure 19

We generate our various modal logic systems with different restrictions on '□' on left and '◇' on right. In the following '*i*' ranges over a sequence of numerals and '*m*', '*n*' range over numerals.

□ on L , ◇ on R out: K	$\Box/\Diamond\phi(\mathbf{i})$	$\phi(\mathbf{i}+\mathbf{m})$ (old)
□ on L , ◇ on R out: D	$\Box/\Diamond\phi(\mathbf{i})$	$\phi(\mathbf{i}+\mathbf{m})$ (new)
□ on L , ◇ on R out: T	$\Box/\Diamond\phi(\mathbf{i})$	$\phi(\mathbf{i})$
□ on L , ◇ on R out: B	$\Box/\Diamond\phi(\mathbf{i}+\mathbf{m})$ old	$\phi(\mathbf{i})$
□ on L , ◇ on R out: 4	$\Box/\Diamond\phi(\mathbf{i})$	$\phi(\mathbf{i}+\mathbf{m}+\ldots+\mathbf{n})$ (old)
□ on L , ◇ on R out: 5	$\Box/\Diamond\phi(\mathbf{i}+\mathbf{m})$ old	$\phi(\mathbf{i}+\mathbf{n})$ (old)

Figure 20

Each system has some combination of □-out-on-left/◇-out-on-right rules:

System	Rules
K	K rule
D	K + D rules
T	K + T rules
B	K + T + B rules
S4	K + T + 4 rules
S5	K + T + B + 4 + 5 + rules

Figure 21

The best way to illustrate this system is to prove the characteristic thesis within each system.

3.3 Proofs of Characteristic Theses

In the following, we use lowercase, italicized Greek letters for arbitrary SL wffs. We append an italicized, bracketed world index to each wff. There will be some comments and arrows on the proof to illustrate which rule we're using but are not part of the proof proper.

K proof:

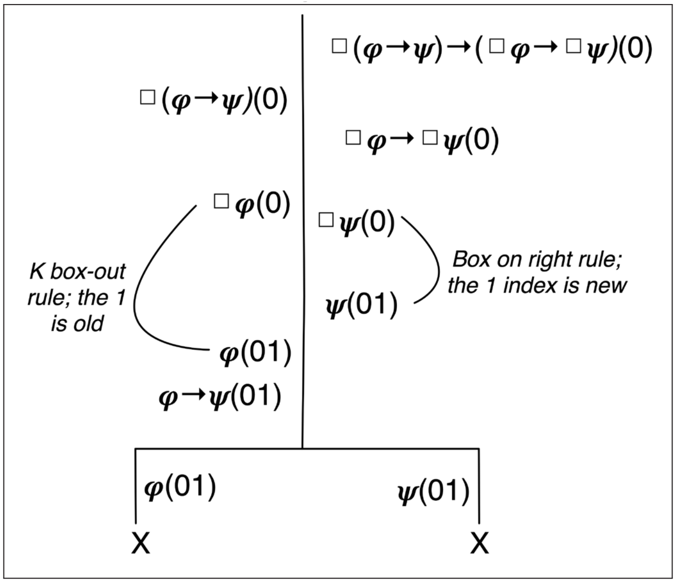


Figure 22

D proof:

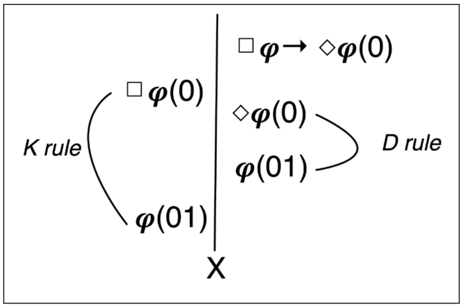


Figure 23

T proof:

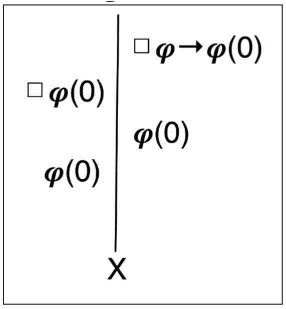


Figure 24

The final $\Box$-out on the left is the T rule.
B proof:

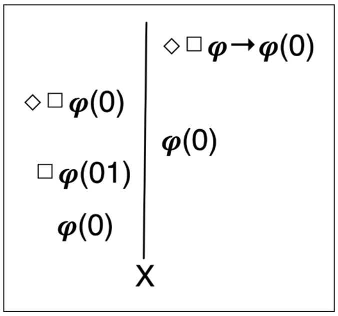


Figure 25

The final $\Box$-out on the left is the B rule.

S4 proof:

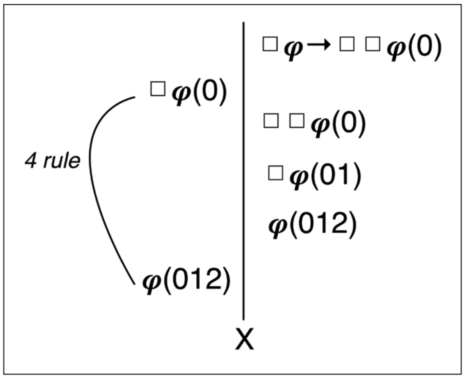


Figure 26

S5 proof:

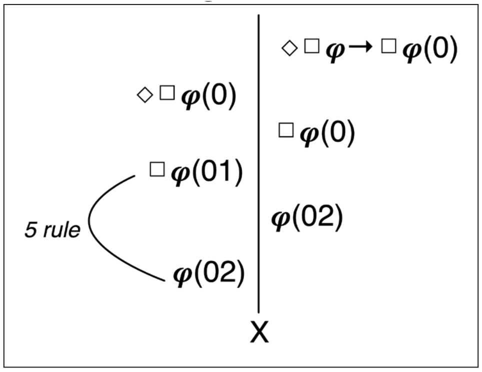


Figure 27

Notes

1. We know of no textbook that employs the system we explore in this paper, although the system is not new: Michael Perloff was teaching the system in the 1990s.

2. Jeffrey refers to them as "truth trees"—with truth being, of course, a semantic notion.

3. See https://courses.umass.edu/phil511-gmh/MAIN/IHome-4.htm.

4. Our system allows for any combination of a set of wffs, a single wff, or the empty set on either, or both, sides of the single turnstile. Any such single turnstile expression is correct just in case the tree closes. Thus, for example, ⌜ Γ ⊢ ⌝, in our system, says that a two-sided tree with the wffs in Γ on the left closes. Some of the semantic analogues are familiar: ⌜ ⊨φ ⌝, where φ is a single wff, says that φ is a tautology. Less familiar, perhaps, are the following. ⌜ ⊨Γ ⌝ says that there is no interpretation in which all the sentences in Γ are false. ⌜ Γ⊨ ⌝ says that Γ is an inconsistent set of wffs. ⌜ φ⊨ ⌝ says that φ is a contradiction.

5. See Bendall 1978: §2.

6. These are often called "nand" and "nor," respectively.

7. Some technicalities. First, officially, a set of indices is admissible iff the set is non-empty; every sequence begins with 0, has a unique terminal element, and numerals aren't skipped; and is closed with respect to predecessor relation so that if 01 is in the set, so is 0, and so on. Second, in the tree, if 0369 had a descendent, that decedent would be 036910, which may seem confusing, but (a) this rarely comes up, and (b) is parsable with close attention to the nature of these sequences.

Bibliography

Bendall, Kent. 1978. "Natural Deduction, Separation, and the Meaning of Logical Operators," *Journal of Philosophical Logic* 7(1): 245–76. https://doi.org/10.1007/BF00245930

Garson, James W. 2006. *Modal Logic for Philosophers*. Cambridge: Cambridge University Press. https://doi.org/10.1017/CBO9780511617737

Girle, Rod. 2000. *Modal Logics and Philosophy*. Kingston, Ont.: McGill-Queen's University Press.

Humberstone, Lloyd. 2000. "The Revival of Rejective Negation," *Journal of Philosophical Logic* 29(4): 331–81. https://doi.org/10.1023/A:1004747920321

Jeffrey, Richard C. 1967. *Formal Logic: Its Scope and Limits*. Indianapolis: Hackett Publishing Co.

Smullyan, Raymond. 1968. *First Order Logic*. Berlin: Springer. https://doi.org/10.1007/978-3-642-86718-7

Jesse J. Fitts is an instructor at the University of Nevada Las Vegas. He recently finished his Ph.D. at the University of Massachusetts Amherst. Professor Fitts works primarily in the philosophy of language, especially propositions, and primarily teaches introductory and advanced logic. E-mail: jesse.fitts@unlv.edu.

David Beisecker is an associate professor at the University of Nevada Las Vegas and former chair of the department. Professor Beisecker specializes in the philosophy of language, philosophical logic, and the classical pragmatist tradition. His most recent research shows how insights from the pragmatist tradition (largely coming from C. S. Peirce) can inform debates in contemporary philosophy of mind, language, and logic. Professor Beisecker holds a Ph.D. from the University of Pittsburgh (1999). E-mail: beiseckd@unlv.nevada.edu.